IN THE SPECIFICATION:

Please replace the paragraph beginning at p. 1, line 9, with the following:

There are many ways in which data can be presented to a user on a user interface. One of the more popular ways is to organize the data into categories and present the categories on one part of the screen. The user may then select a category and in response, the user interface displays data relating to the selected category on another part of the screen. Referring to FIG. 1, for example, a user interface 10 is shown. The user interface 10 includes a left pane 4 and a right pane 6. A disk directory is presented as a set of folder icons 2 in the left pane 4. Each folder icon 2 represents a sub-directory. When the user selects a folder icon, the folder icon is highlighted and files within the sub-directory represented by the selected folder icon are displayed in the right pane 6. In one popular type of categorized presentation, the categories are organized into a graphical hierarchy, such as a tree. Referring to FIG. 2, for example, the folder icons 2 from FIG. 1 are displayed as nodes or "branches" of a tree 12.

A1

Please replace the paragraph beginning at p. 5, line 3, with the following:

Referring to FIG. 5, an example of a computer on which the invention described herein may be implemented is shown. In its most basic configuration, the computer, generally labeled 100, typically includes at least one processing unit 112 and memory 114. Depending on the exact configuration and type of the computer, the memory 114 may be volatile (such as RAM), non-volatile (such as ROM or flash memory) or some combination of the two. This most basic configuration is illustrated in ~~FIG. 2~~ FIG. 5 by

A2

dashed line 106. The computer 100 may also have additional features/functionality. For example, computer 100 may include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 100. Any such computer storage media may be part of computer 100.

A2

Please replace the paragraph beginning at p. 7, line 4, with the following:

When the program that displays the user interface 50 is executed, it does not have any initial knowledge as to what categories are supposed to be listed in the right pane 52. The program obtains this information in a process referred to herein as "populating" the categories or, in the example user interface 50 (FIG. 6), populating the tree 68 with nodes. This process may be fast, or, if there is some problem in retrieving the node data – e.g. the names of the categories are stored on a remote server that has crashed – this process may be very slow. In conventional user interfaces, the user interface simply freezes until the tree can be populated. According to an embodiment of the invention, a

A3

program retrieves the categories independently from displaying them. This technique helps ensure that the user interface does not freeze up with the categories that are being retrieved or "populated." While the categories are being retrieved, the program gives status information to the user concerning the retrieval operation. Referring to FIG. 7, for example, the program displays an incomplete tree 70 on the user interface 50 while the category information is being retrieved. The tree 70 has a placeholder 72 as one of its nodes. The placeholder 72 tells the user the status of the remaining tree nodes. In this case, the remaining tree nodes are being retrieved, as signified by the "retrieving" label. According to an embodiment of the invention, if the program is implemented with independent threads of execution, the retrieving thread may retrieve data for all child nodes if it can be done with a single request to the data storage. Otherwise, the retrieving thread only retrieves direct descendents of the node being populated. This helps eliminate the unnecessary processing of data that is not needed immediately.

Please replace the paragraph beginning at p. 12, line 4, with the following:

When the user selects a computer such as by clicking on a node of the tree 68, the state of the main thread moves to block 240 of FIG. 10. At block 240, the main thread checks the cache 202 (FIG. 9) to see if the disk directory data for the selected computer is there. If the data is there, the main thread moves to block 242, at which it retrieves the data from the cache and displays it in the right panel 54 (FIG. 6). When ~~complete~~completed, the main thread returns to block 220. If the data is not in cache, then the main thread moves to block 238, in which it checks the queue 200 (FIG. 9) to determine if a request for the data has already been entered. If a request for the disk

directory data associated this the selected machine is already in the queue, the main thread returns to block 220. If it is not already in the queue, then the main thread moves to block 236, at which it places a request for the disk directory data in the queue.

A4

Please replace the paragraph beginning at p. 12, line 16, with the following:

As previously described, a user may raise the priority of an in-progress retrieval of disk directory data by activating a pop-up menu, or by re-selecting a previously selected category. The main thread reacts by moving to block 230 (FIG. 10), at which it looks in the queue 200 (FIG. 9) to see if the request that is to be boosted is still there. If it is, then the main thread moves to block 228, at which the it moves the request to the front of the queue. If the request is not in the queue – such as when the request is already being serviced by a worker thread – then the main thread puts a request to have the priority of the data retrieval boosted. For example, if the user indicated that he wanted the priority of the retrieval operation for Machine 1 directory data (FIG. 6), then the main thread would look in the queue to see if the request for Machine 1 data was still pending. If it was, then the main thread would move the request to the front of the queue. If not, then the main thread would put a new request at the front of the queue. The purpose of the new request would tell the control thread 208 to find the worker thread that is currently retrieving the Machine 1 directory data and boost its priority. The main thread would then return to block 220.

A5